

DESCRIÇÃO DE UM SISTEMA DE ENTRADA E SAÍDA PARA UMA
MÁQUINA MULTIPROCESSADORA

J. S. FARIAS^{*} , M. M. MATTOS^{**} , H. M. RODRIGUES^{***}

SUMARIO

Este artigo descreve o sistema de entrada e saída para uma máquina constituída por dois microprocessadores MC68000 desenvolvida no Curso de Pós-graduação em Ciência da Computação da UFRGS.

Apresenta-se de forma detalhada a estrutura do sistema, que permite o atendimento paralelo das requisições de entrada e saída, e seu inter-relacionamento com o núcleo do sistema principal.

ABSTRACT

This paper describes the input/output system for the machine composed by two MC68000 developed at Curso de Pós-graduação em Ciência da Computação da UFRGS.

The paper presents in detail the structure of the system which allows parallel execution of I/O requests, and its relationship with the nucleus of the main system.

*Engenheiro Civil (UFBA,1979), Professor Assistente do Departamento de Ciências Exatas da Universidade Estadual de Feira de Santana (UEFS), mestrando em Ciência da Computação (UFRGS); áreas de interesse: sistemas operacionais e computação gráfica; Curso de Pós-Graduação em Ciência da Computação UFRGS; Av. Osvaldo Aranha, 99 - Porto Alegre - RS - Cep 90210.

**Tecnólogo em Processamento de Dados (UNISINOS, 1984), Professor Auxiliar do Departamento de Computação da UNISINOS (1986); mestrando em Ciência da Computação (UFRGS); áreas de interesse: sistemas operacionais e programação concorrente; Curso de Pós-Graduação em Ciência da Computação UFRGS; Av. Osvaldo Aranha, 99 - Porto Alegre - RS - Cep 90210.

***Bacharel em Matemática (UFV, 1976), Professor Auxiliar do Departamento de Matemática da Universidade Federal de Viçosa (UFV), mestrando em Ciência da Computação (UFRGS); área de interesse: sistemas operacionais. Curso de Pós-Graduação em Ciência da Computação UFRGS; Av. Osvaldo Aranha, 99 - Porto Alegre - RS - Cep 90210.

O desenvolvimento do presente trabalho tem apoio financeiro da SID Informática S/A através do convênio firmado com o CPGCC-UFRGS.

INTRUDUÇÃO

Encontra-se em fase de desenvolvimento no CPGCC da UFRGS um projeto que preve a construção de uma máquina com dois microprocessadores MC68000 e a implementação de um sistema operacional para esta máquina.

Cada microprocessador possui uma memória local e compartilha além de uma memória global um processador de entrada e saída (PES), o qual tem acesso as memórias locais.

Tradicionalmente entrada e saída (E/S) é considerada uma das mais sordidas áreas do projeto de sistemas operacionais. A razão para este fato está na grande variedade de periféricos, cujas características e modo de operação variam enormemente. Especificamente periféricos podem diferir em velocidade, unidade de transferência, representação dos dados, operações permitidas e condições de erro.

Na solução adotada optou-se por implementar atendimento paralelo das requisições de E/S geradas pelos processos, utilizando-se um PES. Assim, pelo fato de toda operação de E/S ser executada via PES, a CPU é liberada para realizar outras tarefas.

A seguir o ambiente do sistema é descrito. Inicialmente é caracterizado o hardware que o suportará e posteriormente sua estrutura e filosofia de funcionamento.

ARQUITETURA DA MÁQUINA

A figura 1 ilustra a arquitetura da máquina, a qual é composta por 2 microprocessadores MC68000, cada um deles podendo acessar uma memória local (particular) de 128 Kb e uma memória global (compartilhada) de igual tamanho.

Os dois microprocessadores estão dispostos em uma placa, a qual está ligada ao PES. Este é um microcomputador ED-281 com a seguinte configuração: CPU 280A, 112 Kbytes de memória, uma unidade de disco rígido tipo winchester, duas unidades de disco flexível de 8" e uma impressora de 100 cps.

A fim de realizar as operações de E/S o PES pode ser interrompido ou interromper cada MC68000, além de poder acessar cada memória local.

INTERFACE NÚCLEO DIAÇAI - PES

Diaçai é um núcleo de multiprogramação em fase de desenvolvimento no CPGCC-UFRGS escrito na linguagem de programação "C" e portanto bastante portátil.

Uma implementação do mesmo em ambiente IBM-PC e compatível com o MS-DOS é apresentada num outro artigo também submetido para apre-

sentação no I SIMPOSIO BRASILEIRO DE ARQUITETURA DE COMPUTADORES E PROCESSAMENTO PARALELO.

Diaçai pode ser integrado a um programa concorrente através de uma biblioteca ou acessado através de interrupções. No primeiro caso, as primitivas são chamadas como funções "C".

O núcleo possui primitivas de sincronização baseadas em variáveis semáforo e primitivas de temporização que permitem o controle de execução de um processo em função de tempo, permitindo assim, entre outros, o desenvolvimento de sistemas de tempo real.

A estratégia round-robin com preempção é utilizada para o escalonamento dos processos, sendo a criação e destruição dos mesmos dinâmica e, realizada através de primitivas específicas. O gerenciamento da memória (alocação, liberação e compartilhamento) utiliza a política first-fit.

Um processo quando executa uma instrução de entrada e saída realiza uma chamada ao núcleo do Sistema Operacional, perdendo consequentemente o processador. Este processo é então inserido na fila de processos bloqueados até a operação de E/S ter sido completada, quando é colocado na fila de processos aptos a novamente concorrer pela CPU. Entretanto, a ação de bloquear os processos não faz parte do procedimento de requisição de entrada e saída.

Validada a requisição, ela pode ser inserida na fila de requisições na memória local do processador em questão, caso o recurso esteja liberado. Caso contrário, ela é inserida na fila associada ao recurso solicitado, mantida pelo Sistema Operacional a fim de garantir acesso exclusivo ao mesmo. Após a liberação do recurso, o primeiro elemento desta fila é inserido na fila de requisições.

Sempre que um elemento é inserido na fila de requisições, uma interrupção é gerada com o objetivo de informar ao PES a existência da mesma.

Completada a operação de E/S, o PES interrompe o processador que a requisitou, causando assim uma chamada do núcleo, que devolve ao processo o status da operação e o coloca na fila ready.

Em cada memória local há uma cópia do núcleo, sendo a memória global utilizada para fins de intercomunicação, sincronização e controle de acesso a recursos compartilhados.

Tendo como base este ambiente de software e hardware, o Sistema Operacional a ser construído poderá distribuir os seus componentes de acordo com as necessidades específicas da aplicação a que se destina.

ESTRUTURA E FUNCIONAMENTO

Em cada memória particular dos microprocessadores MC68000 há uma fila de requisições de E/S e uma fila de operações já executadas

pelo PES (figura 2). A fila possui tantos elementos quanto for o número de periféricos ligados ao PES, sendo que cada elemento possui a seguinte estrutura.

```
type REQUISICAO =  
  record  
    NUMERO_PROCESSO  
    PERIFERICO  
    TICKET  
    OPERACAO  
    ENDERECO_LEITURA  
    ENDERECO_ESCRITA  
    STATUS  
  end;
```

O tamanho da fila está associado ao número de periféricos disponíveis pois o acesso a estes é feito de forma mutuamente exclusiva. Por consequência no PES não é necessário definir uma fila e sim manter um registro descritor da operação associado a cada periférico.

Na memória global há uma variável chamada ticket cujo objetivo é garantir a ordem sequencial de atendimento às requisições. Esta é inicializada com zero e é incrementada pelo núcleo toda vez que uma requisição de E/S for solicitada. A partir do ticket que o processo "toma" ao requisitar uma operação de E/S, o PES verifica qual é a próxima requisição a ser atendida, obtendo seus parâmetros da fila na memória local do processador que causou a interrupção, garantindo assim uma política justa de atendimento.

A inserção de um elemento na fila de requisições é informada ao PES através da execução da seguinte instrução.

```
MOVE $2,$40000      ; carrega no endereço 40000H o comando de  
                    ; interrupção do PES
```

Ela faz com que o hardware transfira o valor armazenado na posição 40000H (registrador de comandos do MC68000) para o registrador de status (E4H) do PES, onde a requisição é indicada pelo nível lógico 1 do bit correspondente ao processador (bit 0 corresponde a P0 e bit 1 corresponde a P1). Esta ação ocasiona uma interrupção IRQ14 que no PES é controlada por dois elementos PIC 8259 (Programmable Interrupt Controller), ligados em uma configuração mestre escravo.

A rotina de atendimento de interrupção, ao ser disparada, consulta o registrador de status e liga uma variável de controle indicando assim ao PES a existência de uma requisição a ser atendida. Então o PES, através de uma rotina específica, transfere os parâmetros da fila na memória local do processador que o interrompeu para o registro de operação associado ao periférico em questão.

O registro de operação apresenta a seguinte estrutura:

```
type REGISTRO_DE_OPERACAO =
```

```

record
PROCESSADOR
OPERACAO
ENDERECO_LEITURA
ENDERECO_ESCRITA
STATUS
ENDERECO_NA_FILA
end;

```

A introdução dos campos PROCESSADOR e ENDERECO_NA_FILA permitem devolver o status da operação após o atendimento da requisição.

O endereçamento para a transferência de dados de/para as memórias locais é feito utilizando-se dois registradores de endereçamento (EOH e ElH), cujas estruturas são mostradas na figura 3. Como as memórias estão divididas em 8k páginas de 16 bytes, são necessários 13 bits para endereçá-las. Estes bits são obtidos através da concatenação dos 5 bits menos significativos do registrador EOH com os 8 bits do registrador ElH. O deslocamento dentro da página, 0H a FH, é obtido a partir dos 4 bits menos significativos das portas de endereço FOH a FFH do PES. O bit 5 do registrador EOH permite selecionar a memória local que se deseja referir.

Realizada a operação de E/S, é devolvida uma indicação do resultado da mesma na fila de operações realizadas (ocupando o mesmo nó em que estava representada a requisição, na estrutura de fila mostrada na figura 2) na memória local do processador que requisitou a operação. Ao término da operação o PES através de uma interrupção, gerada via o registrador de comandos (E2H), indica ao processador requisitante o término da mesma. Isto é feito através das seguintes instruções:

```

LD  A, XH ; carrega o acumulador com o comando desejado
OUT (0E2H),A ; escreve no registrador o comando armazenado
                ; no acumulador

```

onde X pode assumir os valores 1 ou 2 conforme o processador em questão.

Com isso o processador interrompido devolve o status da operação realizada para o processo que a requisitou, colocando o mesmo na fila de processos aptos.

Além da interrupção, o registrador de comandos do PES permite a execução de operações HALT e RESET nos processadores P0 e P1 (figura 4). Isto é feito de maneira semelhante ao procedimento de interrupção, utilizando-se o mesmo código acima descrito, substituindo-se contudo o valor de X, conforme o comando desejado. Assim, para executar uma operação de HALT, X deve assumir os valores 8H ou 4H respectivamente para P0 ou P1. Da mesma forma, para executar a operação RESET, X deve assumir os valores 10H (P1) ou 20H (P0).

O sistema de E/S paralelo é totalmente dirigido por interrupção e é composto basicamente pelos drivers associados a cada periférico disponível, pela rotina que dispara o atendimento de requisições e

pela rotina que periodicamente verifica o status dos periféricos.

Compõe ainda o sistema, as rotinas que realizam a interface com o Sistema Operacional. Elas possibilitam a transferência de requisições para os registros de operação, a devolução do status na fila de operações executadas e a transferência de dados entre os buffers nas memórias locais e os buffers associados a cada periférico.

O paralelismo no atendimento das requisições é obtido modificando-se convenientemente os drivers de E/S do BIOS (Basic Input Output System) de forma a permitir disparar operações de E/S em diversos periféricos simultaneamente.

Assim, os drivers que colocam a CPU em busy-wait aguardando o término da operação de E/S, devem ser modificados de forma que esta possa executar outras tarefas enquanto as operações de E/S propriamente ditas estão sendo realizadas. Contudo estes periféricos necessitam ainda de um monitoramento da CPU a fim de que seja determinado o status da operação. A solução adotada consiste em realizar um polling periódico nos periféricos em questão, com operações de E/S pendentes.

O núcleo do sistema de E/S tem a seguinte estrutura:

```
carga_do_sistema;
repeat
  if variável_controle
  then
    begin
      carrega registro de operação do periférico associado;
      atualiza variável de controle;
      dispara_atendimento_requisição(periférico);
    end;
  devolve_status_operação;
forever;
```

A rotina responsável por efetivamente disparar o atendimento da requisição é a seguir mostrada:

```
procedure dispara_atendimento_requisição(periférico);
begin
  if periférico.operação = escrita
  then
    begin
      transfere conteúdo do buffer na memória local para o buffer
      associado ao periférico;
    end;
  inicializa_atendimento;
end;
```

A partir dos parâmetros constantes no registro de operação ela requisita transferência do conteúdo do buffer na memória local para o buffer do periférico (no caso de se tratar de uma operação de escrita) e inicializa o atendimento propriamente dito.

A rotina `devolve_status_operação` é chamada pelo sistema a fim de devolver o status para a fila de operações executadas e informar ao MC68000 que solicitou atendimento, o término da operação.

```
procedure devolve_status_operação;
begin
  determina_periférico;
  if status_ok
  then
  begin
    if periferico.operação = leitura
    then
    begin
      transfere conteúdo do buffer do periférico para a memória
      local;
    end;
    transfere status da operação para a fila de operações atendi-
    das;
    interrompe o MC68000 que requisitou a operação;
  end;
end;
```

Esta rotina faz uso dos campos `N_PROCESSADOR` e `ENDERECO_NA_FILA`, constantes no registro de operação do periférico em questão.

INICIALIZAÇÃO DO SISTEMA

A inicialização do sistema é realizada a partir do PES. Inicialmente deve ser executado o programa que implementa o sistema de entrada e saída. Este programa dispara o procedimento inicial de carga cuja função é transferir de disco para as memórias locais, em posições físicas pré-definidas, o código que implementa o sistema operacional. Ao término desta transferência, é enviado um sinal de reset para cada processador, fazendo com que os mesmos entrem em estado de processamento. Realizado o procedimento inicial de carga, o PES entra em estado de busy-wait até ser interrompido por algum dos processadores, para atender uma requisição de entrada e saída.

CONCLUSÃO

O trabalho descrito neste artigo faz parte de um projeto em andamento no Curso de Pós-Graduação em Ciência da Computação da UFRGS. A fixação dos componentes na placa, conforme definido no projeto de hardware, já foi concluído e sua ligação ao PES efetuada. Atualmente um sistema de entrada e saída que permite o atendimento sequencial das requisições está em fase de teste.

Tendo em vista que este projeto é um protótipo, que permitirá o futuro desenvolvimento de um sistema mais arrojado, optou-se por soluções simples de forma a tê-lo funcionando no menor tempo possível.

BIBLIOGRAFIA

- /CLA 86/ CLAFF, W.J. Writing assembly language interrupt routines. Byte 249-62, Extra edition, 1986.
- /COS 86/ Costa, C.M. et alli. O sistema de entrada e saída da máquina Micro-Big. XIII SEMISH Anais. Recife-Olinda, julho 1986.
- /JOH 83/ JOHNSON-LAIRD, A. The programmer's CP/M handbook. Osborne McGraw-Hill, Berkeley, California, 1985.
- /LEV 79/ LEVENTHAL, L.A. Z80 Assembly language programming. Osborne McGraw-Hill, Berkeley, California, 1979.

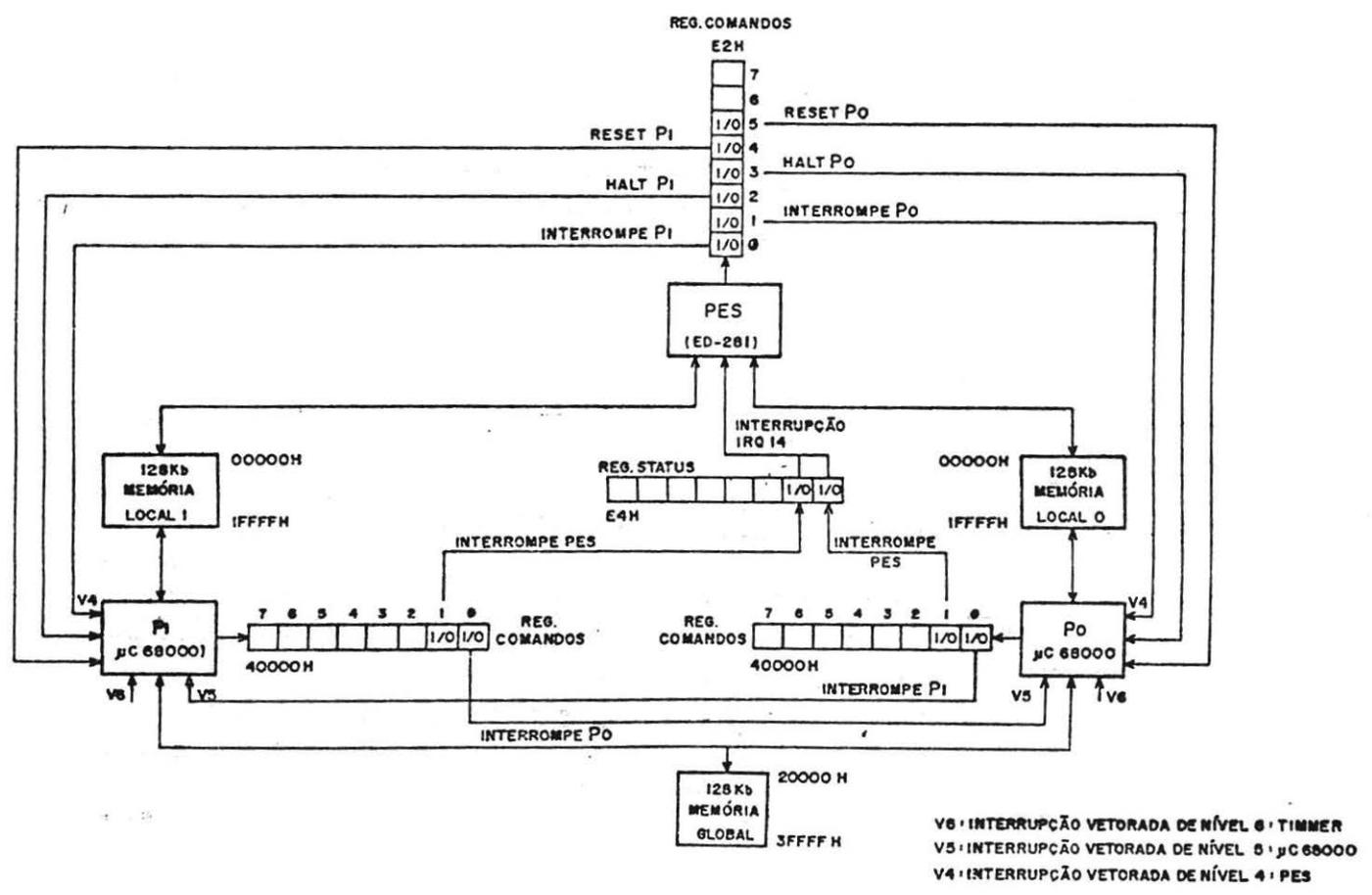
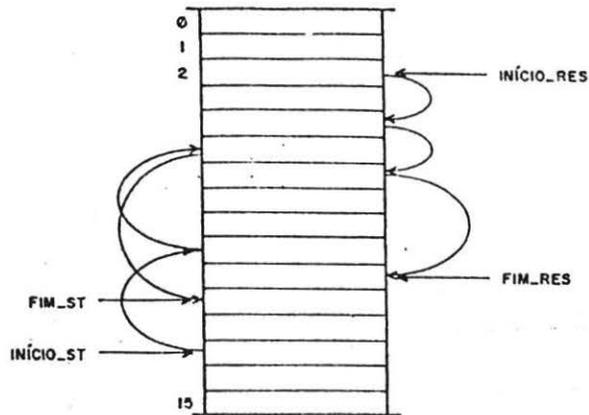


Figura 1 - Configuração Detalhada do Sistema.



INÍCIO-RES: APONTADOR PARA INÍCIO DA FILA DE REQUISIÇÕES DE ENTRADA E SAÍDA;
FIM-RES: APONTADOR PARA FIM DA FILA DE REQUISIÇÕES DE ENTRADA E SAÍDA;
INÍCIO-ST: APONTADOR PARA INÍCIO DA FILA DE STATUS;
FIM-ST: APONTADOR PARA FIM DA FILA DE STATUS.

Figura 2. - Filas de Requisição de E/S e de Status nas memórias locais nos processadores 68000.

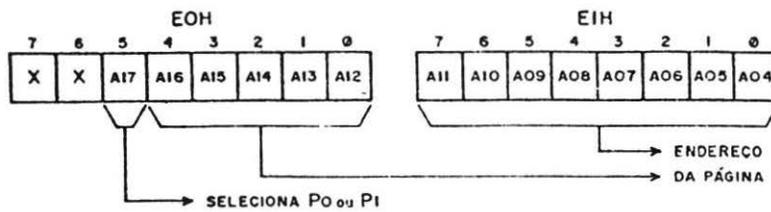
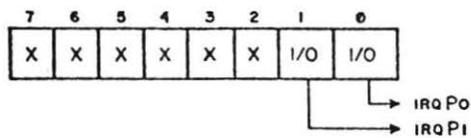


Figura 3- Registradores de Endereçamento.

Registrador de Status : E4H



Registrador de Comandos : E2H

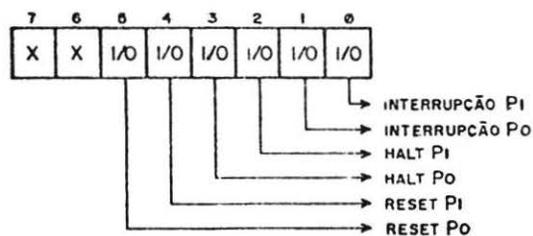


Figura 4- Lay-out dos Registradores de Comandos e Status do PES.